



175 E. Hawthorn Pkwy, Suite 401, Vernon Hills, IL 60061  
Tel: (847) 816-1231 • Fax: (847) 816-0568  
[info@intechnic.com](mailto:info@intechnic.com)

## The Bible of Web Development

Revision: September 1, 2005

The following guidelines are to be followed by all Intechnic Corporation personnel involved in web development. This document is to be used as an every-day reference for Project Managers, Development Managers, Programmers, Coders and Designers on all levels and on all projects. Changes and additions to this document apply to projects and work started after the revision date of this document.

This document outlines a rigid set of standards to be followed in the development process. All of these standards must be followed by default and without any exceptions UNLESS otherwise instructed by Project Managers. Project Managers are responsible for overseeing and enforcing these guidelines on all levels.

THIS DOCUMENT AND ITS CONTENTS IS NOT PUBLIC DOMAIN AND IS PROPERTY OF INTECHNIC CORPORATION. ANY COPYING, TRANSMITTING OR DISTRIBUTING OF THIS DOCUMENT TO ANY PARTIES OUTSIDE OF INTECHNIC CORPORATION OR USING THIS DOCUMENT FOR ANY PURPOSE OTHER THAN THE WORK IN INTECHNIC CORPORATION IS STRICTLY PROHIBITED AND IS IN VIOLATION OF COMPANY'S POLICIES AND INTERNATIONAL LAWS.

### 1. Table of Contents

1.	Table of Contents .....	1
2.	Design & Flash.....	5
2.1.	General.....	5
2.1.1.	Resolution .....	5
2.1.2.	Color Rate.....	5
2.1.3.	Layout.....	5
2.1.4.	Slice-up and Usability Requirements.....	5
2.2.	Flash.....	5
2.2.1.	Loading Indicator .....	5
2.2.2.	Preloading .....	5
2.2.3.	Sound / Music.....	5
2.2.4.	Skip Flash Presentation.....	5
3.	Coding / Slice-Up.....	6
3.1.	Browser Compatibility .....	6

3.2.	Technology Standards .....	7
3.2.1.	XHTML .....	7
3.2.2.	HTML Comments .....	7
3.2.3.	Validation .....	7
3.2.4.	Technology List.....	7
3.3.	Slice-Up .....	7
3.3.1.	Resolution .....	7
3.3.2.	Layout.....	8
3.3.3.	Default Template .....	8
3.3.4.	Spacer.....	8
3.3.5.	Tabbing .....	8
3.4.	CSS.....	8
3.4.1.	Default CSS .....	8
3.4.2.	Default Text.....	8
3.4.3.	CSS Environment .....	8
3.4.4.	CSS Declarations.....	8
3.4.5.	Tag Attributes.....	8
3.4.6.	CSS Class Names .....	9
3.5.	Fonts .....	9
3.5.1.	Default Font Types .....	9
3.5.2.	Default Font Size .....	9
3.6.	Java-Script .....	9
3.6.1.	External Libraries.....	9
3.7.	Images.....	9
3.7.1.	Image Folder.....	9
3.7.2.	ALT Tags .....	9
3.7.3.	Preloading for Mouseover.....	9
3.8.	META Tags .....	9
3.8.1.	General Tags.....	9
3.9.	Structure/Miscellaneous .....	9
3.9.1.	Inclusions .....	9
3.9.2.	User Files.....	9
3.9.3.	Mod-Rewrite.....	10
3.9.4.	Custom URLs.....	10
3.10.	Credits/Copyrights .....	10
3.10.1.	Intechnic Credit .....	10
3.10.2.	Copyrights.....	10
4.	Programming .....	10
4.1.	General/Kernel.....	10
4.1.1.	Redirects .....	11
4.1.2.	404 Pages.....	11
4.1.3.	BASE Tag .....	11
4.1.4.	SSL Usage .....	11
4.1.5.	SSL Links .....	11
4.1.6.	Mixed SSL.....	11
4.1.7.	Dates.....	11
4.2.	Order/Payment Processing.....	11
4.2.1.	Gateway Credentials .....	11

4.2.2.	Expiration Dates .....	11
4.2.3.	Credit Card Numbers.....	11
4.2.4.	Order/Checkout Form .....	12
4.3.	Moving/Setting Up Sites .....	12
4.3.1.	General .....	12
4.3.2.	Permissions .....	12
4.3.3.	CRON .....	12
4.4.	Testing/Debugging.....	12
4.4.1.	Debugging Messages .....	12
4.4.2.	Credit Card Testing .....	12
4.4.3.	Test Records .....	12
4.4.4.	Test Account .....	12
4.5.	Naming Convention .....	12
4.5.1.	General .....	12
4.5.2.	Comments.....	13
4.5.3.	Capitalization Styles.....	13
4.5.4.	.NET Capitalization Rules.....	13
4.5.5.	Case sensitivity .....	13
4.5.6.	Abbreviations .....	14
4.5.7.	Word Choice (.NET Only).....	14
4.5.8.	Name Space Naming Guidelines (.NET Only).....	14
5.	Databases .....	14
5.1.	Database Structure.....	14
5.1.1.	Primary Key .....	14
6.	Usability.....	14
6.1.	Navigation.....	15
6.1.1.	Logo Link.....	15
6.1.2.	Title .....	15
6.1.3.	Consistent Titles.....	15
6.1.4.	Consistent Icons .....	15
6.1.5.	Active Menu .....	15
6.1.6.	Mousover Menu.....	15
6.1.7.	Multi-level Menu.....	15
6.2.	Regional (U.S. Sites) .....	15
6.2.1.	Regional Settings.....	15
6.2.2.	Date Format .....	15
6.2.3.	Time Format.....	15
6.2.4.	Number Format.....	15
6.2.5.	Currency Format.....	15
6.3.	Regional (European Sites) .....	16
6.3.1.	Regional Settings.....	16
6.3.2.	Date Format .....	16
6.3.3.	Time Format.....	16
6.3.4.	Number Format.....	16
6.3.5.	Currency Format.....	16
6.4.	Spelling/Grammar.....	16
6.4.1.	Spell-check .....	16
6.4.2.	Company / Client Names .....	16

6.4.3.	Carryovers.....	16
6.4.4.	Special Case Text .....	16
6.4.5.	Title Case.....	16
6.4.6.	Administrative Console.....	16
6.5.	Forms.....	17
6.5.1.	Consistent Field Labels .....	17
6.5.2.	Positioning and Alignment.....	17
6.5.3.	Trailing Semicolon.....	17
6.5.4.	Fields in Title Case.....	17
6.5.5.	Grouping.....	17
6.5.6.	Pagination .....	17
6.5.7.	Contact Forms.....	18
6.5.8.	Country Field.....	18
6.5.9.	State Field.....	18
6.5.10.	Required Fields.....	17
6.5.11.	Validation .....	17
6.5.12.	E-mail Validation .....	17
6.5.13.	Passwords Validation .....	17
6.5.14.	Field Hints .....	17
6.5.15.	Submit.....	18
6.5.16.	Tabulation.....	18
6.5.17.	Double-Submit.....	18
6.6.	Data Grid .....	18
6.6.1.	Consistency of Table Headers and Fields .....	19
6.6.2.	Default Sorting.....	19
6.6.3.	Filters .....	19
6.6.4.	Totals .....	19
6.6.5.	IDs .....	19
6.6.6.	Column Titles .....	19
6.6.7.	Items Per Page.....	19
6.6.8.	Configuration .....	19
6.7.	Pagination.....	20
6.7.1.	Default Pagination .....	20
6.7.2.	Pagination Usability.....	20
7.	Project Management.....	20
7.1.	General .....	20
7.1.1.	Changes to Code.....	20
7.1.2.	Changes to Templates / Images and Styles .....	20
7.2.	Project Specifications .....	20
7.2.1.	Documented Specifications.....	20
7.2.2.	Definition of Specs.....	20
7.2.3.	Supplementary Materials.....	20
7.2.4.	Missing Specs.....	21
7.2.5.	Following Specs & Initiative .....	21
7.2.6.	Changes to Specs .....	21
8.	Appendix .....	21
8.1.	Browsers/Platforms Share .....	21
8.2.	Common Cross-browser Compatibility Issues.....	22

8.3.	Supported Screen Resolution.....	22
8.4.	Commonly Used Phrases.....	22
8.5.	Commonly Misspelled/Misused Words.....	23
8.6.	Order Statuses .....	24

## 2. Design & Flash

### 2.1. General Requirements

#### 2.1.1. Resolution

All websites must be designed to fit on 800x600 screen resolution.

#### 2.1.2. Color Rate

All websites must be designed to support color rate of 16-bit.

#### 2.1.3. Layout

All websites must feature resizable (expandable) design unless requested otherwise by the Project Manager. In other words, the layout needs to look good on resolutions that are higher than 800x600. Combination method (when certain parts are expandable and certain are fixed) is preferred as generally it is advisable to have content in a fixed-width placeholder.

#### 2.1.4. Slice-up and Usability Requirements

Designers must understand the requirements for slice up/coding and usability and produce design that meets those requirements. It is expected of designers to consult coders in case of uncertainties.

### 2.2. Flash

#### 2.2.1. Loading Indicator

Each flash needs a loading indicator so that the user sees a progress while the flash is loading.

#### 2.2.2. Preloading

All of flash's elements need to be preloaded as not to cause any interruptions when it is played.

#### 2.2.3. Sound / Music

For Flash presentations that are embedded into design, if they contain sound or music, there needs to be a "sound off" button. Site should be written in such a manner so that it remembers the user's setting until cookies expire or are erased.

#### 2.2.4. Skip Flash Presentation

For Flash presentations that automatically open as an intro (full screen or as a pop-up) there should always be an option to skip or close such presentations. It is advisable that intro flash presentations play automatically only to first time users, then cookies get set and the user goes directly to the homepage (that may contain an option to play the flash again)

### 3. Coding / Slice-Up

#### 3.1. Browser Compatibility

Websites designed by Intechnic must work correctly on the platforms and browsers listed in the table below. The slice up prepared by coders must be tested across browsers and platforms for compatibility and the resulting code must reasonably work well on all browsers and platforms. Any large-scale changes and additions performed directly by programmers must also be validated by coders (such validations may be requested at any time if the programmers feel that certain HTML/layout changes may pose problems).

While it may not be possible to make certain sites look identical on all browsers and platforms, it is our goal to achieve that and all critical elements of the site (such as navigation, for example) MUST work on all browsers and platforms listed below.

The only exception is Administrative Console which is in the transitional period until further notice. Administrative Console is currently supported in IE, version 6.0 and FF, versions 1.06 and higher for Windows.

<b>Browsers supported</b>	<b>Platform(s) supported</b>	<b>Site must be tested on</b>	<b>Browser/platform combination usage in USA</b>
1. Internet Explorer (IE) 5.0 and higher	MS Windows 95, 98, Millennium, 2000, XP	IE 6.0 on Windows XP	83%
2. Mozilla Firefox (FF) 1.0 and higher		FF 1.0.4 on Windows XP	5%
3. Netscape Navigator (NN) 7.0 and higher		NN 7.1 on Windows XP	2%
4. Opera (OP) 6.0 and higher		OP 6.0 on Windows XP	1.8%
5. Mozilla (MZ) 1.6 and higher		MZ 1.6 on Windows XP	1%
6. Internet Explorer for Mac (IEM) 5.2 and higher	Mac	IEM 5.2 on Mac	1%
7. Mozilla Firefox (FF) 1.0 and higher		FF 1.0.4 on Mac	1%
8. Safari (SF)		SF 1.0.4 on Mac	2%
9. Mozilla Firefox (FF) 1.0 and higher	Linux	FF 1.0.4 on Linux Fedora Core	3%
10. Opera (OP) 6.0		OP 6.0 on Linux	0.2%

and higher		Fedora Core	
------------	--	-------------	--

### 3.2. Technology Standards

#### 3.2.1. XHTML

For all HTML coding, XHTML version 1.1 must be used. This includes any templates, or HTML output. ([www.w3.org/TR/xhtml1/](http://www.w3.org/TR/xhtml1/))

#### 3.2.2. HTML Comments

Resulting HTML must contain minimum comments. If something needs to be removed, backup the code and remove it, do not leave it commented out in HTML.

#### 3.2.3. Validation

All of the resulting HTML (any output) must pass validation at <http://validator.w3.org/>

#### 3.2.4. Technology List

The following technology usages must be passing the following validators:

Technology	Definition/Version	Validator
1. HTML	HTML 4.0.1 Transitional ( <a href="http://www.w3.org/TR/html4/">www.w3.org/TR/html4/</a> )	<a href="http://validator.w3.org/">http://validator.w3.org/</a>
2. XHTML	XHTML 1.1 ( <a href="http://www.w3.org/TR/xhtml1/">www.w3.org/TR/xhtml1/</a> )	<a href="http://validator.w3.org/">http://validator.w3.org/</a>
3. CSS	CSS 2.0 ( <a href="http://www.w3.org/Style/CSS/">www.w3.org/Style/CSS/</a> )	<a href="http://jigsaw.w3.org/css-validator/">http://jigsaw.w3.org/css-validator/</a>
4. XML	XML 1.1 ( <a href="http://www.w3.org/TR/xhtml11/">www.w3.org/TR/xhtml11/</a> )	<a href="http://www.w3schools.com/dom/dom_validate.asp">http://www.w3schools.com/dom/dom_validate.asp</a>
5. DOM	DOM levels 1, 2, 3 ( <a href="http://www.w3.org/DOM/">www.w3.org/DOM/</a> )	
6. XSLT	XSLT 2.0 ( <a href="http://www.w3.org/TR/xslt20/">http://www.w3.org/TR/xslt20/</a> )	<a href="http://www.w3.org/TR/xslt20/">http://www.w3.org/TR/xslt20/</a>

### 3.3. Slice-Up

#### 3.3.1. Programming and Design Requirements

Coders must understand how the slice up/coding will be used by programmers and produce HTML that meets the requirements of the designers and programmers. It is expected of coders to consult programmers in case of uncertainties.

#### 3.3.2. Resolution

The resulting slice-up must fit on screen resolution of 800x600.

### 3.3.3. Layout

The resulting slice-up must be resizable (expandable) unless requested otherwise. (See 2.1.3 for more details)

### 3.3.4. Default Template

Default template file to be used for any new templates - [index.tpl](#) (file attached)

### 3.3.5. Spacer

1x1 px clear dot should be used as a spacer. It must be named "s.gif".

### 3.3.6. Tabbing

One TAB is equal to 2 spaces (4 spaces may be used for HTML coding). One TAB should be used to outline every element inside.

### 3.3.7. Changes to Programming

Coders must not make any modifications to functional code (such as PHP or .NET). They must communicate them to programmers instead.

## 3.4. CSS

### 3.4.1. Default CSS

Default CSS file is to be used for new sites - [style.css](#). Default styles that must be present in every project are: Default Text, Page Title, Subtitle and Caption. These styles must be designed by designers, coded by coders and automatically included in the CMS Editor (if CMS is present) and must match the original design. The description of each style is the following:

- Default Text – any “regular” body text throughout the website.
- Page Title – the main tile on top of every page (biggest and boldest)
- Subtitle – the secondary title used in the body of the page (bigger and bolder than the regular text)
- Caption – text used for references, captions for images, hints, comments, notes, etc (usually smaller or in italics)

### 3.4.2. Default Text

By default, any new text entered into the CMS must be in style Default Text (matching the default text in the design) and it needs to reflect that in the CMS Editor’s style selector.

### 3.4.3. CSS Environment

The styles/environment displayed in the CMS Editor must match the styles/environment on the front end (and obviously the styles in the original design). For example, if we have white text (Arial, 10pt) on blue background on the front end, we must have white text (Arial, 10pt) on the blue background in the CMS editor.

### 3.4.4. CSS Declarations

All CSS declarations should be external (file named "style.css" located in "templates/inc/"). Do not declare CSS styles in the templates.

### 3.4.5. Tag Attributes

Avoid using tag attributes in HTML (such as width, height, bgcolor, color, cellpadding, cellspacing, etc.) and use CSS styles as much as possible instead (declared externally).



#### 3.4.6. CSS Class Names

CSS Class names should consist of Latin names, numbers and "-" (no underscores "\_"). Class name should give a clear understanding of class usage purpose.

### 3.5. Fonts

#### 3.5.1. Default Font Types

Arial, Verdana or Helvetica should always be used unless a specific font is requested.

#### 3.5.2. Default Font Size

Default font size: "s", 10pt or 11px, unless a specific font is requested. Resizable font sizing style ("s") is preferred.

### 3.6. Java-Script

#### 3.6.1. External Libraries

JavaScript code (functions and definitions) should always be done in external .js files located in "templates/inc/", instead of including it directly in HTML. The default (main file) should always be called script.js. Additional files can be created in the same location, as needed. The important thing is not to leave JavaScript code in the front-end template files (HTML) and keep as much of it as possible in external files.

### 3.7. Images

#### 3.7.1. Image Folder

All images used in the layout and design (for template formations) must be located under "templates/img/".

#### 3.7.2. ALT Tags

All "functional" or "important" images must have ALT tags in them. For example: logotypes (name of the company); diagrams (explaining what's on the diagram), text done as images (slogans, captions, design elements), menu items, imagemaps, etc.

#### 3.7.3. Preloading for Mouseover

All menu items and other features that have mouseover effects need to have the mouse-over images preloaded when the page initially loads.

### 3.8. META Tags

#### 3.8.1. General Tags

Each page must have the following META tags: Keywords, Description and Author. Keywords and Description need to be connected to the Structure Management System (if there is one). Author tag needs to say "Intechnic Corporation".

### 3.9. Structure/Miscellaneous

#### 3.9.1. Inclusions

All file inclusions other than the folders described above should be done in the folder "templates/inc".

#### 3.9.2. User Files

All of user files uploaded through CMS Editor (images/documents, etc.) must be located at "templates/UserFiles" and not in "templates/img".

### 3.9.3. Mod-Rewrite

All sites need have mod-rewrite installed by default.

### 3.9.4. Custom URLs

Each site must be prepared to change/rename URLs without significant effort. It is advised that only virtual pages (pages without functionality) are used for SEO purposes.

## 3.10. Credits/Copyrights

### 3.10.1. Intechnic Credit

Each site needs to have Intechnic's credits in the bottom right corner (right aligned) of every page. Credits are done in the format of a text "Powered by Intechnic", followed by the Intechnic's logo on the next line. Both are links to <http://www.intechnic.com/> and the image has an ALT tag "Intechnic Corporation". If the logo link can not be placed, place a text link "Powered by Intechnic". All of the design mock-ups produced by Intechnic must already have credits placed in them.

### 3.10.2. Copyrights

At the left side of the footer, each site needs to have the copyright message. Default copyright message "©[Year] Company Name. All Rights Reserved". [Year] should always be dynamically replaced by the current year. Optionally, above or next to the copyright message, other "footer" links can be placed, such as: Terms and Conditions of Use, Terms and Conditions of Sale, Privacy Policy, Site Map, etc.

## 4. Programming

### 4.1. Technology Standards

The following technology standards are in effect on Intechnic's servers as of 2005/09/01. All of new development work must be designed for and tested on the following versions, unless otherwise instructed and confirmed by the Project Manager AND the Hosting Department.

#### 4.1.1. Linux (LAMP)

The following LAMP (**L**inux **A**pache **M**ySQL **P**HP) configuration is the platform of choice for all web development projects and is to be used by default unless otherwise instructed by Project Manager.

- Apache 1.3.x
- PHP 4.4.x
- MySQL 4.1.x
- Perl 5.8.5

#### 4.1.2. Windows (shared)

- OS: Win2000
- SQL: 2000
- .NET Framework: 1.1
- ColdFusion MX

#### 4.1.3. Windows (dedicated - recommended)

- OS: Win2003 Standard (Web optional)

## 4.2. General/Kernel

### 4.2.1. Redirects

Any internal or external redirects performed within Kernel should be permanent, i.e. response type 301. PHP code for this is: `header("HTTP/1.1 301 Moved Permanently");`  
`header("Location: $destination");`

### 4.2.2. 404 Pages

For every project, we need to create a 404 “System Page” for Not Found Requests. If CMS and SMS is available, there must be a separate template for it. Kernel must also return the proper response type with this page: 404. PHP code for this is:  
`header("HTTP/1.1 404 Not Found");`

### 4.2.3. BASE Tag

Eliminate using BASE tag in all front-end templates. We don't really need it and if this tag is formatted incorrectly, it can cause a lot of problems for browsers and SE spiders.

### 4.2.4. SSL Usage

SSL (HTTPS links) should be used only when they are necessary - login pages, restricted areas, checkout pages, admin, etc. Avoid or disallow using HTTPS for public content pages.

### 4.2.5. SSL Links

Links to non-secure pages must always be “http://” (non-secure); links to secure pages must always be “https://” (secure). For example, on a secure page, all links to non-secure pages must be “http://” and not “https://”.

### 4.2.6. Mixed SSL

Make sure that there are no pages that contain a mix of secure and non-secure elements (such as images). Nowhere on the site front end or administrative console should browser give a warning saying that the user is loading a page with “potentially insecure items”.

### 4.2.7. Dates

Whenever programming anything that has to do with submission of birth dates or any other dates that have a possibility of a date earlier than January 1<sup>st</sup>, 1970, check that the date is stored correctly in the database (i.e. use Date Format, not Int to store it in the database)

## 4.3. Order/Payment Processing

### 4.3.1. Gateway Credentials

If the website is using a credit card processing feature and if we are using a payment gateway, the username, the password, the transaction key (or it's equivalent if Authorize.Net is not used), and the gateway name should be in the configuration section of the site.

### 4.3.2. Expiration Dates

When creating forms for credit card processing, make sure that the expiration year dropdown shows current year + 7 years from current year.

### 4.3.3. Credit Card Numbers

Any time a credit card number is entered, it must have basic validation. The rules are: digits only; 16 digits for Visa, MasterCard and Discover; 15 digits for American Express. The actual numbers start at: 4 for Visa, 5 for MasterCard, 6 for Discover and 3 for American Express.

#### 4.3.4. Order/Checkout Form

The order/check out form has to be protected against double-submit (double click), re-submit (using the Back button) or interrupting (Stop button). It must be designed in a way that browser's Stop, Refresh and Back buttons won't interfere with the processing (when a request is sent to the payment gateway but the response is not received, because it was interrupted by the user).

### 4.4. **Moving/Setting Up Sites**

#### 4.4.1. General

When moving or setting up new websites, programmers are responsible for making sure that the site is operational. This includes but not limited to verifying forms and special functionality, file uploads and permissions, CRON, etc.

#### 4.4.2. Permissions

For every new site (or when the site is moved) file permissions need to be set and checked for any user and administrative image/file upload functionality.

#### 4.4.3. CRON

If the site requires CRON to operate, it must be set-up and tested by programmers.

### 4.5. **Testing/Debugging**

#### 4.5.1. Debugging Messages

Any visible debugging messages/output can not be performed on the live site unless specifically allowed by the Project Manager. If it is absolutely necessary, they must be visible in source code only (as HTML comments).

#### 4.5.2. Credit Card Testing

All credit card processing features need to be tested on the live site with a real credit card with the processing gateway set to live mode at least once before the work is classified as completed. All test transactions should be made for \$1 unless otherwise required by the PM or by the types of purchases that need to be tested.

#### 4.5.3. Test Records

When testing on the live site, any test records/users/orders, etc. must be deleted off the live site as soon as possible. If testing requires a long term/permanent creation of a record, approval from the Project Manager must be received. Such records must contain "intechnic test" or "intechtest" to indicate that it is a test record created by Intechnic. Also, the amount of such records should be kept to a minimum. (U.S Only) All test records cannot contain LV information. LV e-mail address should also not be used (use .com instead)

#### 4.5.4. Test Account

There should be ONE test account on live sites. Each developer cannot have his/her own unless this is needed for testing purposes. The test account should always be "intechnic" with the default password (from Development Manager)

### 4.6. **Naming Convention**

#### 4.6.1. General

The naming convention rules apply to both .Net and PHP code (where applicable).

#### 4.6.2. Comments

The code must be reasonably commented to document features and applications that are not obvious.

#### 4.6.3. Capitalization Styles

Use either Pascal Case, Camel Case or UpperCase capitalization style depending on where you use it

#### 4.6.4. .NET Capitalization Rules

(.Net only) The following table summarizes the capitalization rules and provides examples for the different types of identifiers.

Identifier	Case	Example
Class	Pascal	<b>AppDomain</b>
Enum type	Pascal	<b>ErrorLevel</b>
Enum values	Pascal	<b>FatalError</b>
Event	Pascal	<b>ValueChanged</b>
Exception class	Pascal	<b>WebException</b> <b>Note</b> Always ends with the suffix <b>Exception</b> .
Read-only Static field	Pascal	<b>RedValue</b>
Interface	Pascal	<b>IDisposable</b> <b>Note</b> Always begins with the prefix <b>I</b> .
Method	Pascal	<b>ToString</b>
Namespace	Pascal	<b>System.Drawing</b>
Parameter	Camel	<b>typeName</b>
Property	Pascal	<b>BackColor</b>
Protected instance field	Camel	<b>redValue</b> <b>Note</b> Rarely used. A property is preferable to using a protected instance field.
Public instance field	Pascal	<b>RedValue</b> <b>Note</b> Rarely used. A property is preferable to using a public instance field.

#### 4.6.5. Case sensitivity

Do not use names that require case sensitivity. Components must be fully usable from both case-sensitive and case-insensitive languages.

Misuse:

```
void MyFunction(string a, string A)
```

Proper use:

```
void MyFunction(string a, string b)
```

#### 4.6.6. Abbreviations

To avoid confusion and guarantee cross-language interoperability, follow these rules regarding the use of abbreviations:

- Do not use abbreviations or contractions as parts of identifier names. For example, use `GetWindow` instead of `GetWin`.
- Do not use acronyms that are not generally accepted in the computing field.
- Where appropriate, use well-known acronyms to replace lengthy phrase names. For example, use `UI` for `User Interface` and `OLAP` for `On-line Analytical Processing`.
- When using acronyms, use Pascal case or camel case for acronyms more than two characters long. For example, use `HtmlButton` or `htmlButton`. However, you should capitalize acronyms that consist of only two characters, such as `System.IO` instead of `System.Io`.
- Do not use abbreviations in identifiers or parameter names. If you must use abbreviations, use camel case for abbreviations that consist of more than two characters, even if this contradicts the standard abbreviation of the word

#### 4.6.7. Word Choice (.NET Only)

Avoid using class and variable names that duplicate commonly used .NET Framework namespaces or other languages commonly used keywords, for example `System`, `Collections`, `Forms` ....

Avoid language specific terminology. Use names that describe a type's meaning rather than names that describe the type.

#### 4.6.8. Name Space Naming Guidelines (.NET Only)

The general rule for naming namespaces is to use the company name followed by the technology name and optionally the feature and design as follows.

```
CompanyName.TechnologyName[.Feature][.Design]
```

Prefixing namespace names with a company name or other well-established brand avoids the possibility of two published namespaces having the same name. Use a stable, recognized technology name at the second level of a hierarchical name. Use organizational hierarchies as the basis for namespace hierarchies.

## 5. Databases

### 5.1. Database Structure

#### 5.1.1. Primary Key

All tables in the database should have a primary index field specified at all times. All tables should be optimized for index usage before releasing site to live.

#### 5.1.2. Indexing

For all queries that query more than one table (left join), make sure that the reference fields are indexed. Proper indexation sometimes increases performance by as much as 500%.

## 5.2. Queries

### 5.2.1. Nested Queries

Avoid using nested sub-queries if they repeat in a large query more than once. The code will work much faster if the repeating sub-query is executed first and the results are fed into the primary query.

## 6. Usability

### 6.1. Navigation

#### 6.1.1. Logo Link

The company logo must always be the link to the home page

#### 6.1.2. Title

Every page of the site must have a title displayed at the top of the page

#### 6.1.3. Consistent Titles

The title of the page must always match the page name used in the navigation menu; unless otherwise requested by Project Manager.

#### 6.1.4. Consistent Icons

If icons are used in the navigation menu and in the title, icons in the title must match with those in the navigation menu.

#### 6.1.5. Active Menu

The navigation menu must always display the actively selected section.

#### 6.1.6. Mousover Menu

The navigation menu must have a mouseover effect.

#### 6.1.7. Multi-level Menu

If the navigation menu is multi-level, each level must be visually different from the parent (usually by indentation and smaller font)

### 6.2. Regional (U.S. Sites)

#### 6.2.1. Regional Settings

Programmers and DMs are responsible for setting correct regional settings for U.S. sites.

#### 6.2.2. Date Format

Default date format for U.S. sites is mm/dd/yyyy.

#### 6.2.3. Time Format

Default time format for U.S. sites is 12-hour (example: 11:05 PM).

#### 6.2.4. Number Format

For U.S. sites thousands are separated by commas (“,”). Point (“.”) is used for fractions (for example: 1,000.00).

#### 6.2.5. Currency Format

All figures that represent amounts (money) must be properly recoded in the currency format. Default currency format for U.S. sites is \$X.XX (example: \$1,000.00). Notice that “\$” is placed before the number.

### **6.3. Regional (European Sites)**

#### **6.3.1. Regional Settings**

Programmers and DMs are responsible for setting correct regional settings for European sites.

#### **6.3.2. Date Format**

For European sites it is dd.mm.yyyy.

#### **6.3.3. Time Format**

Default time format for European sites is 24-hour (example: 23:05).

#### **6.3.4. Number Format**

For European sites, fractions are indicated by commas (“,”) (for example: 1000,00)

#### **6.3.5. Currency Format**

All figures that represent amounts (money) must be properly recoded in the currency format. In Europe, the currency symbol is placed after the number (example: 1000,00 USD)

### **6.4. Spelling/Grammar**

#### **6.4.1. Spell-check**

Any text (copy) placed on the site must be spellchecked first (unless that text was provided by PM, in which case they have to spell-check it). Any person making spelling/grammar mistakes on the site without spellchecking will be held responsible.

#### **6.4.2. Company / Client Names**

Company and client names need to be specially checked to make sure that there are no typos or mistakes. Also, attention needs to be paid to case and spacing of letters.

#### **6.4.3. Carryovers**

When copying anything from a different site special attention must be paid to make sure that there are no “carryovers”, things proprietary to the other site: like other company’s name, site name, etc.

#### **6.4.4. Special Case Text**

All of special case text that is not clearly visible or is conditionally displayed (such as pop-up messages, prompts, error messages, etc) needs to be approved by the Project Manager before being placed on the live site (unless specified in this document). DM is responsible for notifying the PM about new error messages (because they are not clearly visible) and PM is responsible for providing such text..

#### **6.4.5. Title Case**

(U.S only) All items in the navigation menu, page titles and subtitles must always be done in the Title Case (<http://www.sitepoint.com/blog-post-view.php?id=243299>). There are no exceptions to this rule.

#### **6.4.6. Administrative Console**

The proper name is Administrative Console. We must always spell out “Administrator” or “Administrative Console” and never use “admin” on the actual sites. “Admin” can only be used in communications between developers but not as a term on the web site.



## 6.5. Forms

### 6.5.1. Consistent Field Labels

All field names must be consistent throughout the site, including admin. For example, if there is a field (“First Name:”) on the front end, then the same field must be labeled (“First Name:”) in Administrative Console.

### 6.5.2. Positioning and Alignment

The field label must always be to the left of the field. All fields must be spaced evenly and aligned properly on the left and on the right.

### 6.5.3. Trailing Semicolon

All form labels must have a trailing “:” (example: “First Name:”)

### 6.5.4. Fields in Title Case

All form labels must be in the title case (See 6.4.5)

### 6.5.5. Grouping

Fields on large forms need to be separated into logical groups (for example: General Information, Contact Information, Payment Information, etc.)

### 6.5.6. Pagination

Forms that do not fit on 3 printed out pages and require excessive scrolling must be split up into multiple steps (by logical grouping). In such case, each step must be clearly labeled, numbered and displayed in the page title. User must be able to move from page to page by the means of “Back” and “Next” buttons on the form and in the browser. The validation must be done on every step before proceeding to the next step.

### 6.5.7. Required Fields

Each form that has required fields must say “\* Required Fields” at the top. Each required field must be marked with red “\*” right after the field label with a space before it (example: “First Name: \*”)

### 6.5.8. Validation

If a required field is missing or invalid, an error message needs to be displayed (in red) at the top of the form: “One or more fields on this form have an error. Please review the fields marked with red carefully and submit the form again.” The field’s label needs to be highlighted in red and the first offensive field needs to be focused. To the right of the offensive field we need to show the message: “This field is required” or “This e-mail address is invalid” (etc.)

The design created for forms must always include room for validation (error messages) and sample error message (including styles) should be included with the form.

### 6.5.9. E-mail Validation

Always validate e-mail addresses that they are in the format [xxx@xxx.xxx](#)

### 6.5.10. Passwords Validation

Always include a second field for the password (Password Again) and check that they match.

### 6.5.11. Field Hints

For certain fields that have certain requirements, we need to place a hint to the right of the field. For example: for Password “(min. 5 characters)”. The hint has to be in a smaller font than the field label.

6.5.12. Submit

Pressing Enter when filling out a form should submit it.

6.5.13. Tabulation

Pressing Tab when filling out a form should jump to the next field on the form.

6.5.14. Double-Submit

All forms must be protected against double-submit (double click on the Submit button) for submitting the same results twice.

6.5.15. Refreshing

If the form refreshes for any reason (loading additional choices depending on a selection, for example) it must preserve the values in the filled out fields.

## 6.6. Form Fields Usage

6.6.1. Contact Forms

(U.S only) Forms containing contact information should always be done in this format (and in this order):

First name – text field (full width)  
Last name – text field (full width)  
Street address – text field (full width)  
Street address (cont.) – text field (full width)  
City – text field (full width)  
State – standard dropdown (full width)  
Zip – text field (5 characters)  
Country – standard dropdown (full width)

6.6.2. Country Field

Country field should always be a dropdown with the standard list of countries (link) in the alphabetical order. (U.S only) United States should always come first and be automatically selected.

6.6.3. State Field

(U.S only) If the United States is selected for country, the state field must always be a dropdown with the standard list of states ([http://dev1.intechnic.com/mantis/file\\_download.php?file\\_id=1438&type=bug](http://dev1.intechnic.com/mantis/file_download.php?file_id=1438&type=bug)) in the alphabetical order.  
If the Canada is selected, the state field should be a dropdown of Canadian provinces.

6.6.4. Text Box vs. Text Area

Use a text box if the entry can be reasonably recorded on one line (as one sentence), for example: First Name. Use a text area if there is a possibility that it may be more than one sentence (several lines of text or a paragraph), for example: Comments. Make sure that the size of the text area is adequate to how much text you can expect to be entered. In order for a text area to be effective for working with a lot of text it should be at least 5-6 lines in height and 50-60 characters in width.

6.6.5. Radio Buttons vs. Dropdown

If it is a fixed number of choices (they don't change) and there are less than 3-4 choices, use radio buttons. For example: Enabled/ Disabled, On/Off, etc. Use radio buttons for

multiple choice answers. Always use dropdowns if there a lot of choices or if they can dynamically change. For example: States, User Relation, Parent Category, etc.

#### 6.6.6. Multi-select Dropdown

If you allow multiple selections in your dropdown, make its height at least 5-6 lines.

#### 6.6.7. Radio Buttons vs. Checkboxes

On a standalone form when you have a group of radio buttons or checkboxes, radio buttons and checkboxes should always come first and then the label.

Do no recreate the functionality of radio buttons with checkboxes (un-checking one checkbox if another one is selected). Always use radio buttons instead.

### **6.7. Data Grid**

#### 6.7.1. Consistency of Table Headers and Fields

All field names in the data grid (headers in the actual table) must be consistent with the field names throughout the site, but without the “:”. For example, if there is a field (“First name”) on the front end, then the same field must be labeled (“First name”) in the data grid.

#### 6.7.2. Default Sorting

There should always be a default sorting in each data grid. If not certain, sort by ID, descending (latest on top).

#### 6.7.3. Filters

Filters must be in place for each column in the data grid. For variables that have a fixed list of values, use dropdowns (statuses, for example). For numerical fields and dates, use ranges (from: and to:). Filters must be automatically applied (form submitted) when pressing Enter.

#### 6.7.4. Totals

All of numerical fields (columns) must be automatically added up with the totals displayed at the bottom of the data grid (except for IDs)

#### 6.7.5. IDs

Each newly created data grid must always by default show record IDs. IDs must be unique and must no be reused.

#### 6.7.6. Column Titles

Column titles should always fit on a single line and be lined up with the columns in the data grid (align to the left).

#### 6.7.7. Items Per Page

Each data grid must have a corresponding “Items Per Page” variable in Configuration that needs to be identical with the data grid’s title.

#### 6.7.8. Configuration

Each site’s Configuration section must be manually checked to make sure that all correct variables and values are in place and that there are none missing or transferred from other web sites.

## 6.8. Pagination

### 6.8.1. Default Pagination

Any list that has a variable length (meaning that new records can be added) needs to have pagination. Pagination mechanism must support unlimited number of pages. This also includes any front-end features.

### 6.8.2. Pagination Usability

Pagination should have the active page show in negative colors and it should not be a link. We always need to have “<<” and “>>” to jump 10 pages (displayed if there are previous and next 10-page blocks). We also need to have “|<” and “>|” to jump to the first page and the last page. The pagination needs to look like this:

Page: << |< 10 20 30 40 50 60 70 80 90 >| >>

We are only displaying page numbers for pages that exist. If there is no page, there shouldn't be a number for it in the pagination tool.

## 7. Project Management

### 7.1. General

#### 7.1.1. Changes to Code

Project Managers are not allowed to make ANY modifications to PHP or .NET code. If you need a change, you post it into Mantis for the Development team (not Coders) . If it is an emergency, you must contact VPP.

#### 7.1.2. Changes to Templates / Images and Styles

Templates, styles and images can be updated directly by Project Managers. If they are updated, PM must post a notification (item of type Information) to Mantis about the change that was made.

### 7.2. Project Specifications

#### 7.2.1. Documented Specifications

All explanations, instructions and decisions reached over the phone, e-mail, instant messenger or any other method must be documented by PMs in Mantis. PMs are not allowed to give, and DMs and programmers are not allowed to accept instructions or specifications that are given not through Mantis. DMs and programmers are not responsible for following non-documented specs and instructions and PMs will be held responsible in disputes arising from such situations.

#### 7.2.2. Definition of Specs

Specifications provided by PMs must be in a format of a requirements document, describing how features need to work, what is it they are supposed to do and how they will be used (what are the requirements). PMs should not post any instructions on how to program (how to write the code or how to structure the database).

#### 7.2.3. Supplementary Materials

If PM uploads any supplementary materials (documents, screenshots, etc.), they must upload them to Mantis or post a link to the files in Mantis (so that they files can be easily retrieved over the Web without searching for them). When posting a link to the documents, make sure that it works and that the supplementary materials are actually uploaded/attached.

#### 7.2.4. Missing Specs

If required to meet a deadline, and the DMs does not have full information or a complete set of requirements (missing specs), they may make assumptions on how it is supposed to be done based on the experience from previous projects. In such case, they must describe in Mantis in details how it was done and how it works.

#### 7.2.5. Following Specs & Initiative

DMs, programmers, coders and designers can not do anything that CONTRADICTS the documented specs, or the guidelines in this document unless specifically requested by PMs to do otherwise by documented instructions or updated specs in Mantis. Should DMs, programmers, coders or designers have questions or reservations about the specs, they must post a question/clarification if the time permits and await further instructions. If the time does not permit (pressing deadline) they must do the work according to the specifications and then they may post their questions/suggestions along with the completed work.

#### 7.2.6. Changes to Specs

If specifications are changing, PMs must post all changes to Mantis as soon as possible (to avoid unnecessary work). PMs should not delete old specs from Mantis, they must upload new specs as a separate document/bug note and notify DM of the changes by sending them a reminder. All changes to the original specs must be recorded into the same item in Mantis.

## 8. Appendix

### 8.1. Browsers/Platforms Share

Platform market shares in 2005 (based on [www.macobserver.com](http://www.macobserver.com) data - [article 1](#), [article 2](#)):

	Platform	Market share in USA	
1.	MS Windows 95, 98, NT, Millennium, 2000, XP	90%	
2.	Linux and Unix based OS	6%	
3.	Mac OS	4%	

Browser market shares in 2005 (based on [BrowserNews](#) data):

	Browser	Market share in USA	
1.	Internet Explorer versions 5.0 - 6.0	84%	
2.	Mozilla Firefox 1.0	8%	
3.	Netscape Navigator 5.0 - 8.0	2%	
4.	Safari	2%	
5.	Opera 7.0 - 8.0	2%	
6.	Mozilla	1%	
7.	Internet Explorer 4.0	0.5%	

8.	Netscape Navigator 4.0	0.5%	
----	------------------------	------	--

### 8.2. Common Cross-browser Compatibility Issues

	Browser combination	Web concept or technology	Compatibility problem	Recommended solution
1	IE vs. all other browsers	Layers over Flash	Layers over Flash work correctly only in IE. They are not supported by FF and other browsers.	Use image instead of Flash object for all non IE browsers to preserve the menu functionality.

### 8.3. Supported Screen Resolution

Resolution usage stats in 2005 in USA based on [BrowserNews](#): less than 800x600 - 1%; 800x600 - 27%; 1024x768 and higher - 70%; non standard resolutions on devices - 3%.)

### 8.4. Commonly Used Phrases

Correct Phrase
The information has been sent/submitted/updated.
The record has been added/updated/deleted.
Please log in to access/view/update this page/record.
Your credit card has been charged.
The order has been shipped/processed/approved.
Forgot your password?
This field is required/invalid
Passwords don't match
E-mail address is not valid

### 8.5. Commonly Misspelled/Misused Words

Correct	NOT Correct
<b>Administrative Console</b>	“Administration Console”, “Administration/ive Panel” “Admin”
E-mail	“email”
Charged	“chaged”
<b>Internet</b>	“internet”
<b>Business</b>	“Bussines”, “Busines”
<b>Community</b>	“Comunity”
Submission	“Submition”, “Submision”
Submitted	“Submitted”
Field	“Feild”, “Flid”
Cancelled	“Canceled”
<b>Address</b>	“Adress”, “Addres”, “Adres”
URL	“url”, “Url”
General	“Genreal”
Description	“Decsription”
<b>FedEx</b>	“Fedex”, “FedEX”
<b>eBay</b>	“ebay”, “Ebay”
Label	“lable”
<b>Gray (color)</b>	“grey”

## 8.6. Order Statuses

Correct Status Name & Colors
Pending
Approved
Declined
Incomplete
Deleted
Shipped
Refunded